



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

50

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/902,810	07/10/2001	Anders Hejlsberg	MS1-865US	6062
22801	7590	05/26/2005	EXAMINER	
LEE & HAYES PLLC 421 W RIVERSIDE AVENUE SUITE 500 SPOKANE, WA 99201			RAMPURIA, SATISH	
			ART UNIT	PAPER NUMBER
			2191	

DATE MAILED: 05/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/902,810

Applicant(s)

HEJLSBERG ET AL.

Examiner

Satish S. Rampuria

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 16 February 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-27 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-127 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 05/08/03, 8/22/03, 2/25/04, 9/13/04, 12/17/04, 2/14/06
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

Art Unit: 2191

Response to Amendment

1. This action is in response to the amendment received on 02/16/2005.
2. The objection to specification (hyperlinks and the word ushered) is withdrawn in view of applicant's amendment.
3. Claims amended by the applicant: 1, 16, 19, 22, and 25.
4. Claims pending in the application: 1-27.

Information Disclosure Statement

5. An initialed and dated copy of Applicant's IDS form 1449 filed on 05/08/2003, 8/22/2003, 2/25/2004, 9/13/2004, 12/17/2004, and 2/16/2005 are attached to the instant Office action.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

- (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.
- (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
- (c) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 4-11 is rejected under 35 U.S.C. 102(e) as being anticipated by .NET Framework Essentials, published in 2001 by Lam (hereinafter called Lam).

Art Unit: 2191

Per claim 4:

Lam discloses:

- An XmlReader class of an application program interface, embodied on one or more computer readable media that enables non-cached forward only access to XML data (page 4 of 8, “section 5.5.2.5 XmlReader-The XmlReader object... is a fast, nocached, forward-only way of accessing streamed XML data”), the XmlReader class comprising:
- an XmlReader constructor that enables initialization of a new instance of the XmlReader class (page 5 of 8, “section 5.5.2.5 XmlReader-... XMLTextReader object with the stream of data from the string... is loop through and perform a Read() operation until there is nothing else to read”); and
- a Read method that enables reading of nodes of the XML data via the XmlReader class instance (page 5 of 8, “section 5.5.2.5 XmlReader-... While reading, only when we come across a node of type XmlElement and a node named Order do we start to process the order”).

Per claim 5:

Lam discloses:

- a BaseURI property that identifies a base URI of a current node of the XML data (page 2 of 3, “section 4.2.1 Distributed Hello Server-...Once created channel object... register... object... RemotingConfiguration... calling RegisterWellKnownServiceType() method... pass in the class name, a URI, and a object-activation mode... URI is important... key element... client application will use to refer specifically to this registered object”); and

Art Unit: 2191

- a NodeType property that identifies the type of the current node (page 2 of 8, “section 5.5.2.1 XmlNode and its derivatives-... XmlNode is a base class that represents a single node in the XML document”)

Per claim 6:

Lam discloses:

- An XPathNavigator class of an application program interface, embodied on one or more computer readable media, that enables read-only random access to a data store (page 1 of 8, “section 5.5.1 Xml parser- ... parser... reads the data, notifying the application of the tag or text”), the XPathNavigator class comprising:
- an XPathNavigator constructor that enables initialization of a new instance of the XPathNavigator class (page 1 of 8, “section 5.5.1 Xml parser- ... parser... reads the data, notifying the application of the tag or text”);
- a MoveToFirst method that enables moving to a first sibling of a current node of XML data (page 3 of 8, “section 5.5.2.1 XmlNode and its derivatives-... XmlNode... supports... properties... aid in navigation... XML document... FirstChild”);
- a MoveToNext method that enables moving to a next sibling of the current node (page 3 of 8, “section 5.5.2.1 XmlNode and its derivatives-... XmlNode... supports... properties... aid in navigation... XML document... NextSibling”);
- a MoveToPrevious method that enables moving to a previous sibling of the current node (page 3 of 8, “section 5.5.2.1 XmlNode and its derivatives-... XmlNode... supports... properties... aid in navigation... XML document... PreviousSibling”);

Art Unit: 2191

- a MoveToFirstChild method that enables moving to a first child of the current node (page 3 of 8, “section 5.5.2.1 XmlNode and its derivatives-... XmlNode... supports... properties... aid in navigation... XML document... FirstChild”);
- a MoveToParent method that enables moving to a parent of the current node (page 3 of 8, “section 5.5.2.1 XmlNode and its derivatives-... XmlNode... supports... properties... aid in navigation... XML document... ParenNode”); and
- a NodeType property that enables obtaining the type of the node that is moved to (page 3 of 8, “section 5.5.2.1 XmlNode and its derivatives-... XmlNode... supports... properties... aid in navigation... XML document... use the ChildNode property to navigate down from the root of the tree”).

Per claim 7:

Lam discloses:

- An XslTransform class of an application program interface, embodied on one or more computer readable media, that enables transforming of XML data using an XSLT stylesheet (page 6 of 8, “section 5.5.2.7 XslTransform-XslTransform converts XML from one format to another... XslTransform xslt = new XslTransform()”), the XslTransform class comprising:
- an XslTransform constructor that enables initialization of a new instance of the XslTransform class (page 6 of 8, “section 5.5.2.7 XslTransform-... XslTransform xslt = new XslTransform()”);

Art Unit: 2191

- a Load method that enables loading of the XSLT stylesheet (page 6 of 8, “section 5.5.2.7 XslTransform-... xslt.Load("XSLTemplate.xsl"); and
- a Transform method that enables transforming of the specified XML data using the loaded XSLT stylesheet and outputs the results (page 6 of 8, “section 5.5.2.7 XslTransform-... “xslt.Transform(xDoc, null, writer)”).

Per claim 8:

Lam discloses:

- A set of XmlSchema classes of an application program interface, embodied on one or more computer readable media, that enable constructing and editing of schemas (page 3 of 9, “section 5.3.1.2 XML and Tables Sets-... WriteXMLSchema() dumps on the schema of the tables, including all table sand relationships between tables”), the set of XmlSchema classes comprising:
 - a Schema class that contains a definition of a schema (page 1 of 1, “section 1.1-... “a class can be expressed as an XML Schema Definition (XSD)”);
 - a SchemaObject class that enables creating of an empty schema (page 1 of 1, “section 1.1-... “an object can be converted to and from an XML buffer; a method can be specified using an XML format called Web Services Description Language (WSDL);”); and
- a SchemaCollection class that contains a cache of defined XML Schema Definition language (XSD) and XML-Data Reduced Language (XDR) schemas (page 5 of 5, “section 2.3.4 Interoperability Support-... The XML schema definition tool can also

take a .NET assembly and generate an XML schema definition (XSD) that represents the types within the .NET assembly”).

Per claim 9:

Lam discloses:

- An XmlResolver class of an application program interface, embodied on one or more computer readable media, that enables resolving of external XML resources named by a Uniform Resource Identifier (URI) (page 2 of 3, “section 4.2.1 Distributed Hello Server-...Once created channel object... supports channel registration and object resolution... register... object... RemotingConfiguration... calling RegisterWellKnownServiceType() method... pass in the class name, a URI, and a object-activation mode... URI is important... key element... client application will use to refer specifically to this registered object”), the XmlResolver class comprising:
 - a ResolveURI method that enables resolving the absolute URI from a base URI and a relative URI (page 2 of 3, “section 4.2.1 Distributed Hello Server-...Once created channel object... supports channel registration and object resolution”); and
 - a GetEntity method that enables mapping of the resolved URI to an object containing identified resource (page 2 of 3, “section 4.2.2 Remote Hello Client-... invoke remote method... first activate the remote object and obtain associated proxy on the client side... call GetObject() method of the activator class”).

Art Unit: 2191

Per claim 10:

Lam discloses:

- An XmlDataDocument class of an application program interface, embodied on one or more computer readable media, that enables structured data to be stored, retrieved, and manipulated through a relational dataset (page 7 of 8, “section 5.5.2.8 XmlDataDocument-one of the most important... DataSet with XML... manipulate in memory”), the XmlDataDocument class comprising:
 - a DataSet property that enables obtaining of a dataset that provides a relational representation of the data in a document (page 7 of 8, “section 5.5.2.8 XmlDataDocument-... once we have a DataSet, we can persist the data inside the DataSet into an XML string or file”);
 - a Load method that enables loading of the document using a specified data source and synchronizing the dataset with the loaded data (page 7 of 8, “section 5.5.2.8 XmlDataDocument-... The schema of the DataSet can be loaded and saved as XML Schema Definition (XSD)... XmlDataDocument can be associated with DataSet”).

Per claim 11:

Lam discloses:

- An XmlWriter class of an application program interface, embodied on one or more computer readable media, that enables a non-cached forward only way of generating streams and files containing XML data (page 5 of 8, “section 5.5.2.6 XmlWriter-

Art Unit: 2191

TheXMLWriter object... is a fast, non cached way of writing streamed XML data”), the XmlWriter class comprising:

- an XmlWriter constructor that enables initialization of a new instance of the XmlWriter class (page 5 of 8, “section 5.5.2.6 XmlWriter-... XmlWriter supports namespaces by providing a number of overloaded functions that take a namespace to associate with the element”); and
- an WriteState property that enables obtaining of the state of an instance of the XmlWriter class (page 5 of 8, “section 5.5.2.6 XmlWriter-... writer.WriteStartElement("ElementName", "myns")”); and
- a plurality of Write methods that enable writing XML data via the instance of the XmlWriter class (page 5 of 8, “section 5.5.2.6 XmlWriter-... “XmlTextWriter writer... writer.Close()”).

8. Claim 12 is rejected under 35 U.S.C. 102(e) as being anticipated by US Publication No. 2002/0169679 to Neumayer (hereinafter called Neumayer).

Per claim 12:

Neumayer discloses:

- An XmlValidatingReader class of an application program interface, embodied on one or more computer readable media, that enables DTD, XDR and XSD schema validation (page 2 and 3, paragraph 33 “the incoming data is validated... accomplished through a class called XMLValidator, which is a helper class to check if the XML data is valid”), the XmlValidatingReader class comprising:

Art Unit: 2191

- a ValidationType property that enables obtaining an indication of what type of validation to perform on a document (page 3 and 4, paragraph 33 “A method of validateXML can be used to check the given XML against the schema”);
- a Read method that enables reading of nodes of the document so that validation of the document can be performed (page 3 and 4, paragraph 34 “an MLProcessor class, which extracts information from the XML and creates the rule object”).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 1, 16, 19, 22, and 25 rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 6,418,448 to Sarkar (hereinafter called Sarkar) in view of US Patent No. 6,209,124 to Vermeire et al. (hereinafter called Vermeire) and further in view of Foody et al. (hereinafter called Foody).

Per claim 1:

Sarkar discloses:

- A software architecture for a distributed computing system comprising: an application configured to handle requests submitted by remote devices over a network (col. 5, lines 62-63 “transactions could be triggered through thin client windows communicating persistently with remote databases”).

Sarkar does not explicitly disclose an application program interface to present functions used by the application to access network and computing resources of the distributed computing system, wherein the application program interface comprises a set of classes that make available standards-based support for processing XML documents.

However, Vermeire discloses in an analogous computer system an application program interface to present functions used by the application to access network and computing resources of the distributed computing system (col. 10, lines 54-56 “the operator interface and the application model which allows for actual processing of the particular language source code”), wherein the application program interface comprises a set of classes that make available standards-based support for processing XML documents (col. 20, lines 8-10 “invokes the parser to convert the source code into a tree form, provide editing for the tree and generating the metadata XML using the XML Writer class”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using application program interface to generate the code as taught by Vermeire into the method of receiving the data via remote devices as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the art would be motivated to use an application program interface to provide the use of the same data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Art Unit: 2191

Neither Sarkar nor Vermeire disclose wherein the set of classes are grouped in the application program interface into a plurality of namespaces, and wherein a first of the plurality of namespaces contains classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) and a second of the plurality of namespaces contains an Xpath parser and evaluation engine.

However, Foody discloses in an analogous computer system wherein the set of classes are grouped in the application program interface into a plurality of namespaces (col. 10-11, lines 65-66 and 1-14 “NameSpaces is the information to describe classes... Class Description Framework is provided to enable this capability, and to enable OSAs to override built-in functionality... Class Description Framework consists of a suite of classes which describe: classes, instances, properties, functions (including methods), arguments, and exceptions...”), and wherein a first of the plurality of namespaces contains classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) and a second of the plurality of namespaces contains an Xpath parser and evaluation engine (col. 11, lines 53-67 “...information... constructed... as NameSpaces (XSLT would be inherent in this process since it consist of implementing the XML) are enumerated... would typically use subclasses of each... classes... set information...” also see FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of having plurality of classes into namespaces used by application program interface as taught by Foody into the method of receiving the data via remote devices as taught by the combination of Sarkar and Vermeire. The modification would be obvious because of one of ordinary skill in the art would be motivated to

have plurality of classes into namespaces used by application program interface to provide a single system enabling software objects from multiple heterogeneous object systems to interoperate bi-directionally and to combine larger systems as suggested by Foody (col. 6, lines 52-64).

Per claim 16:

Sarkar discloses:

- one or more applications configured to be executed on one or more computing devices, the applications handling requests submitted from remote computing devices (col. 5, lines 62-63 “transactions could be triggered through thin client windows communicating persistently with remote databases”).

Sarkar does not explicitly disclose a networking platform to support the one or more applications; and an application programming interface to interface the one or more applications with the networking platform, wherein the application program interface comprises a set of classes that make available standards-based support for processing documents written in a markup language.

However, Vermeire discloses in an analogous computer system a networking platform to support the one or more applications (col. 10, lines 54-56 “the operator interface and the application model which allows for actual processing of the particular language source code”); and an application programming interface to interface the one or more applications with the networking platform (col. 10, lines 54-56 “the operator interface and the application model

which allows for actual processing of the particular language source code”), wherein the application program interface comprises a set of classes that make available standards-based support for processing documents written in a markup language (col. 20, lines 8-10 “invokes the parser to convert the source code into a tree form, provide editing for the tree and generating the metadata XML using the XML Writer class”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using application program interface to generate the code as taught by Vermeire into the method of receiving the data via remote devices as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the art would be motivated to use an application program interface to provide the use of the same data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Neither Sarkar nor Vermeire disclose wherein the set of classes are grouped in the application program interface into a plurality of namespaces, and wherein a first of the plurality of namespaces contains classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) and a second of the plurality of namespaces contains an Xpath parser and evaluation engine.

However, Foody discloses in an analogous computer system wherein the set of classes are grouped in the application program interface into a plurality of namespaces (col. 10-11, lines 65-66 and 1-14 “NameSpaces is the information to describe classes... Class Description Framework is provided to enable this capability, and to enable OSAs to override built-in functionality... Class Description Framework consists of a suite of classes which describe:

Art Unit: 2191

classes, instances, properties, functions (including methods), arguments, and exceptions...”), and wherein a first of the plurality of namespaces contains classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) and a second of the plurality of namespaces contains an Xpath parser and evaluation engine, and a third of the plurality of namespaces contains classes used to serialize objects into XML format documents or streams (col. 11, lines 53-67 “...information... constructed... as NameSpaces (XSLT would be inherent in this process since it consist of implementing the XML) are enumerated... would typically use subclasses of each... classes... set information...” also see FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of having plurality of classes into namespaces used by application program interface as taught by Foody into the method of receiving the data via remote devices as taught by the combination of Sarkar and Vermeire. The modification would be obvious because of one of ordinary skill in the art would be motivated to have plurality of classes into namespaces used by application program interface to provide a single system enabling software objects from multiple heterogeneous object systems to interoperate bi-directionally and to combine larger systems as suggested by Foody (col. 6, lines 52-64).

Per claim 19:

Sarkar discloses:

- A computer system including one or more microprocessors and one or more software programs(col. 5, lines 62-63 “transactions could be triggered through thin client windows communicating persistently with remote databases”), the one or more software programs utilizing an application program interface to request services from an operating system(col. 5, lines 62-63 “transactions could be triggered through thin client windows communicating persistently with remote databases”).

Sarkar does not explicitly disclose the application program interface including separate commands to request services that make available support for processing XML documents.

However, Vermeire discloses in an analogous computer system the application program interface including separate commands to request services that make available support for processing XML documents (col. 20, lines 8-10 “invokes the parser to convert the source code into a tree form, provide editing for the tree and generating the metadata XML using the XML Writer class”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using application program interface to generate the code as taught by Vermeire into the method of receiving the data via remote devices as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the art would be motivated to use an application program interface to provide the use of the same data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Neither Sarkar nor Vermeire disclose the separate commands being grouped into different namespaces a first namespace to support XSLT (Extensible Stylesheet Language Transformations) and a second namespace to serialize objects into XML format documents or streams.

However, Foody discloses in an analogous computer system the separate commands being grouped into different namespaces a first namespace to support XSLT (Extensible Stylesheet Language Transformations) (col. 10-11, lines 65-66 and 1-14 “NameSpaces is the information to describe classes... Class Description Framework is provided to enable this capability, and to enable OSAs to override built-in functionality... Class Description Framework consists of a suite of classes which describe: classes, instances, properties, functions (including methods), arguments, and exceptions...” and a second namespace to serialize objects into XML format documents or streams (col. 11, lines 53-67 “...information... constructed... as NameSpaces (XSLT would be inherent in this process since it consist of implementing the XML) are enumerated... would typically use subclasses of each... classes... set information...” also see FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of having plurality of classes into namespaces used by application program interface as taught by Foody into the method of receiving the data via remote devices as taught by the combination of Sarkar and Vermeire. The modification would be obvious because of one of ordinary skill in the art would be motivated to have plurality of classes into namespaces used by application program interface to provide a

Art Unit: 2191

single system enabling software objects from multiple heterogeneous object systems to interoperate bi-directionally and to combine larger systems as suggested by Foody (col. 6, lines 52-64).

Per claim 22:

Sarkar discloses:

- A method comprising: receiving one or more calls from one or more remote devices over a network (col. 5, lines 62-63 “transactions could be triggered through thin client windows communicating persistently with remote databases”), performing the requested XML document processing (col. 5, lines 62-63 “transactions could be triggered through thin client windows communicating persistently with remote databases”).

Sarkar does not explicitly disclose wherein the one or more calls are to one or more functions that make available support for processing XML documents; and performing the requested XML document processing.

However, Vermeire discloses in an analogous computer system wherein the one or more calls are to one or more functions that make available support for processing XML documents; and performing the requested XML document processing (col. 20, lines 8-10 “invokes the parser to convert the source code into a tree form, provide editing for the tree and generating the metadata XML using the XML Writer class”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using application program interface to

Art Unit: 2191

generate the code as taught by Vermeire into the method of receiving the data via remote devices as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the art would be motivated to use an application program interface to provide the use of the same data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Neither Sarkar nor Vermeire disclose the one or more functions being grouped into a plurality of namespaces with a first namespace containing an Xpath parser and evaluation engine and a second namespace containing classes used to serialize objects into XML format documents or streams.

However, Foody discloses in an analogous computer system disclose the one or more functions being grouped into a plurality of namespaces with a first namespace containing an Xpath parser and evaluation engine (col. 10-11, lines 65-66 and 1-14 “NameSpaces is the information to describe classes... Class Description Framework is provided to enable this capability, and to enable OSAs to override built-in functionality... Class Description Framework consists of a suite of classes which describe: classes, instances, properties, functions (including methods), arguments, and exceptions...” and a second namespace containing classes used to serialize objects into XML format documents or streams (col. 11, lines 53-67 “...information... constructed... as NameSpaces (XSLT would be inherent in this process since it consist of implementing the XML) are enumerated... would typically use subclasses of each... classes... set information...” also see FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of having plurality of classes into namespaces

Art Unit: 2191

used by application program interface as taught by Foody into the method of receiving the data via remote devices as taught by the combination of Sarkar and Vermeire. The modification would be obvious because of one of ordinary skill in the art would be motivated to have plurality of classes into namespaces used by application program interface to provide a single system enabling software objects from multiple heterogeneous object systems to interoperate bi-directionally and to combine larger systems as suggested by Foody (col. 6, lines 52-64).

Per claim 25:

Sarkar discloses:

- A method comprising: calling, to one or more remote devices over a network (col. 5, lines 62-63 “transactions could be triggered through thin client windows communicating persistently with remote databases”),
- receiving, from the one or more remote devices, a response to the calling (col. 5, lines 62-63 “transactions could be triggered through thin client windows communicating persistently with remote databases”).

Sarkar does not explicitly disclose one or more functions that make available support for processing XML documents; receiving, from the one or more remote devices, a response to the calling.

However, Vermeire discloses in an analogous computer system one or more functions that make available support for processing XML documents; receiving, from the one or more remote devices, a response to the calling (col. 20, lines 8-10 “invokes the parser to convert the

source code into a tree form, provide editing for the tree and generating the metadata XML using the XML Writer class”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using application program interface to generate the code as taught by Vermeire into the method of receiving the data via remote devices as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the art would be motivated to use an application program interface to provide the use of the same data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Neither Sarkar nor Vermeire disclose the one or more functions being grouped into a plurality of namespaces with a first namespace containing classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) and a second namespace containing classes used to serialize objects into XML format documents or streams.

However, Foody discloses in an analogous computer system the one or more functions being grouped into a plurality of namespaces with a first namespace containing classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) (col. 10-11, lines 65-66 and 1-14 “NameSpaces is the information to describe classes... Class Description Framework is provided to enable this capability, and to enable OSAs to override built-in functionality... Class Description Framework consists of a suite of classes which describe: classes, instances, properties, functions (including methods), arguments, and exceptions...” and a second namespace containing classes used to serialize objects into XML format documents or streams (col. 11, lines 53-67 “...information... constructed... as NameSpaces

(XSLT would be inherent in this process since it consist of implementing the XML) are enumerated... would typically use subclasses of each... classes... set information...” also see FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of having plurality of classes into namespaces used by application program interface as taught by Foody into the method of receiving the data via remote devices as taught by the combination of Sarkar and Vermeire. The modification would be obvious because of one of ordinary skill in the art would be motivated to have plurality of classes into namespaces used by application program interface to provide a single system enabling software objects from multiple heterogeneous object systems to interoperate bi-directionally and to combine larger systems as suggested by Foody (col. 6, lines 52-64).

11. Claims 2, 17, 20, 23, and 26 rejected under 35 U.S.C. 103(a) as being unpatentable over Sarkar, in view of Vermeire, Foody and further in view of .NET Framework Essentials, published in 2001 by Lam (hereinafter called Lam).

Per claims 2, 17, 20, 23, and 26:

The rejection of claim 1 is incorporated, and further, neither Sarkar nor Vermeire or Foody explicitly discloses an XmlReader class that enables non-cached forward only access to XML data; an XPathNavigator class that enables read-only random access to a data store; an XslTransform class that enables transforming of XML data using an XSLT stylesheet; a plurality of Xml Schema classes that enable constructing and editing of schemas; an XmlResolver class that enables resolving of external XML resources named by a Uniform Resource Identifier

(URI); an XmlDocument class that enables structured data to be stored, retrieved, and manipulated through a relational dataset; and an XmlWriter class that enables a non-cached forward only way of generating streams and files containing XML data.

However, Lam discloses in an analogous computer system an XmlReader class that enables non-cached forward only access to XML data (page 4 of 8, “section 5.5.2.5 XmlReader- The XmlReader object... is a fast, nocached, forward-only way of accessing streamed XML data”); an XPathNavigator class that enables read-only random access to a data store (page 1 of 8, “section 5.5.1 Xml parser- ... parser... reads the data, notifying the application of the tag or text”); an XsltTransform class that enables transforming of XML data using an XSLT stylesheet (page 6 of 8, “section 5.5.2.7 XsltTransform-XsltTransform converts XML from one format to another... XsltTransform xslt = new XsltTransform()”); a plurality of Xml Schema classes that enable constructing and editing of schemas (page 3 of 9, “section 5.3.1.2 XML and Tables Sets- ... WriteXMLSchema() dumps on the schema of the tables, including all table sand relationships between tables”); an XmlResolver class that enables resolving of external XML resources named by a Uniform Resource Identifier (URI) (page 2 of 3, “section 4.2.1 Distributed Hello Server- ...Once created channel object... register... object... RemotingConfiguration... calling RegisterWellKnownServiceType() method... pass in the class name, a URI, and a object-activation mode... URI is important... key element... client application will use to refer specifically to this registered object”); XmlDocument class that enables structured data to be stored, retrieved, and manipulated through a relational dataset (page 7 of 8, “section 5.5.2.8 XmlDocument-one of the most important... DataSet with XML... manipulate in memory”); and an XmlWriter class that enables a non-cached forward only way of generating streams and

files containing XML data (page 5 of 8, “section 5.5.2.6 XmlWriter-TheXMLWriter object... is a fast, non cached way of writing streamed XML data”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the classes as described above as taught by Lam into the system of creating the XML document as taught by the combination system of Sarkar nor Vermeire or Foody. The modification would be obvious because of one of ordinary skill in the art would be motivated to use the classes as described above to build high performance, scalable solutions for e-commerce, also, allows the applications to run on different platform through the use of XML as suggested by Lam (page 1 of 1, “section 5.6 Summary”).

12. Claim 3, 13-15, 18, 21, 24, and 27 rejected under 35 U.S.C. 103(a) as being unpatentable over Sarkar, in view of Vermeire, Foody, Lam and further in view of US Publication No. 2002/0169679 to Neumayer (hereinafter called Neumayer).

Per claims 3, 13-15, 18, 21, 24, and 27:

The rejection of claim 2 is incorporated, and further, neither Sarkar nor Vermeire nor Foody nor Lam explicitly discloses an XmlValidatingReader class that enables DTD, XDR and XSD schema validation.

However, Neumayer discloses in an analogous computer system an XmlValidatingReader class that enables DTD, XDR and XSD schema validation (page 2 and 3, paragraph 33 “the incoming data is validated... accomplished through a class called XMLValidator, which is a helper class to check if the XML data is valid”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of validating XML data via a XML validator as taught by Neumayer into the method of creating a XML document via several XML classes as taught in combination system by Sarkar nor Vermeire nor Foody nor Lam. The modification would be obvious because of one of ordinary skill in the art would be motivated to validate the XML to provide secure transactions in business as suggested by Neumayer (page 1, paragraph 4).

Response to Arguments

13. Applicant's arguments with respect to claim 12 have been considered but they are not persuasive.

In the remarks, the applicant has argued that:

- (i) Applicant stated that the rejection of claims 4-12 which should be 4-11 are not anticipated by Lam due to the invention was conceived before the publication date (June 2001) of Lam and declaration under 37 C.F.R. §1.131 is being prepared with supporting documents.
- (ii) Neumayer does not disclose the XmlValidatingReader class of an application program interface, ValidationType property that enables obtaining and indication of what type of validation to perform on a document as recited in claim 12.

Examiner's response:

- (i) Applicant has not filed a declaration under 37 C.F.R §1.131 to over come the date of reference Lam used. Accordingly, Lam reference is proper prior art and remained. Therefore, the rejection is proper and maintained herein.
- (ii) Regarding the XmlValidatingReader class of an application program interface, Neumayer system does provide an XMLValidatingReader class which validates the XML data, it would inherent to read before the validation (see the rejection above). For the ValidationType property that enables obtaining and indication of what type of validation to perform on a document, Neumayer system does provide validation on a XML data, then creates a rule object for further processing where rule class has a RuleID which indicates the properties of the object (see page 3, paragraphs 34-36). Applicant only makes general allegations and does not point out any errors in the rejection. Therefore, the rejection is proper and maintained herein.

14. Applicant's arguments with respect to claims 1, 16, 19, 22, and 25 has been considered but are moot in view of new ground(s) of rejection.

Conclusion

15. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO**

Art Unit: 2191

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Satish S. Rampuria** whose telephone number is **(571) 272-3732**. The examiner can normally be reached on **8:30 am to 5:00 pm** Monday to Friday except every other Friday and federal holidays. Any inquiry of a general nature or relating to the status of this application should be directed to the **TC 2100 Group receptionist: 571-272-2100**

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Tuan Q. Dam** can be reached on **(571) 272-3695**. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Satish S. Rampuria
Patent Examiner
Art Unit 2191
05/31/2005



ANIL KHATRI
PRIMARY EXAMINER